Implicit B-Spline Surface Reconstruction

Mohammad Rouhani, Angel D. Sappa, *Senior Member, IEEE,* and Edmond Boyer

***Abstract*—This paper presents a fast and flexible curve/surface reconstruction technique based on implicit b-spline. This representation does not require any parameterization and it is locally supported. This fact has been exploited in this paper to propose a reconstruction technique through solving a sparse system of equations. This method is further accelerated to reduce the dimension to the active control lattice. Moreover, the surface smoothness and user interaction are allowed for controlling the surface. Finally, a novel weighting technique has been introduced in order to blend small patches and smooth them in the overlapping regions. The whole framework is very fast and efficient and can handle large cloud of points with low computational cost. The experimental results show the flexibility and accuracy of the proposed algorithm to describe objects with complex topologies. Comparisons with other fitting methods highlight the superiority of the proposed approach in the presence of noise and missing data.**

***Index Terms*—Surface Reconstruction, Implicit Representation, Implicit B-Splines, Partition of Unity, Patch-based Representation, 3L Algorithm, Additional Constraints.**

I. Introduction

SURFACE reconstruction is one of the common fundamental problems in computer vision, graphics and CAD aiming at providing a smooth surface to describe the given point cloud [1]. Thanks to the recent development in 3D scanning technology, multi-view geometry and low cost depth sensors such as Kinect, we are provided with large data volume to be analyzed, which can be noisy or corrupted by outliers and missing data. Therefore, having an efficient point cloud representation is the first step for further processing. Triangular mesh, for instance, is one of the simplest ways to process and visualize 3D cloud of points, but it demands huge amounts of memory space for finer representations. Parametric representations, such as NURBS and Bézier surface, deliver smooth representations through their control points, but they suffer from the parameterization problem.

Implicit functions provide smooth and flexible surface representations without requiring any parameterization. They can describe objects with complex geometries through their zero sets and then, visualize them with efficient methods such as Marching Cubes algorithm. Implicit Polynomial (IP), Radial Basis Function (RBF) and Implicit B-Spline (IBS) are some examples of common implicit functions. They have been widely used for computer vision applications such as: range image reconstruction [2], pose estimation [3], point sets registration [4], [5], [6], [7], shape description [8], position invariant object recognition [9], 3D image segmentation [10] and shape transformation [11] to mention a few.

M. Rouhani and E. Boyer are with Morpheo, INRIA Rhone-Alpes, Grenoble 38000, France (e-mail: {mohammad.rouhani, edmond.boyer}@inria.fr); A. D. Sappa is with the Computer Vision Center, Barcelona, Spain, 08193. E-mail: angel.sappa@cvc.uab.es

Fig. 1. IBS patches are reconstructed in small cells and blended together: $(left)$ 3D point cloud of *Buddha*; $(middle)$ the overlapping IBS patches; $(right)$ the blended IBS surface.

In the current paper, implicit B-splines are used to model the solution space for object representation. These functions are constructed from locally supported functions that enable local control, but unlike the parametric B-splines, they avoid parameterization problems. First, we propose the 3L-IBS algorithm to provide the IBS coefficients through a fast algebraic fitting method [12]. Then, we show how the obtained IBS surface can be further controlled by a proper regularization term and additional constraints. The proposed method is computationally efficient since all the formulations are based on the least squares forms, which lead to sparse systems of equations thanks to the locally supported basis functions of IBS.

Fig. 1 illustrates how the IBS patches describe the details of *Buddha* [13]. In this example 219 mid-size IBS patches are reconstructed (all in less than 6 seconds) for describing more than $145k$ points. In the current work, a novel weighting function for partition of unity is also presented in order to blend these local IBS patches such that the IBS patches are only blended in the overlapping regions and remain the same elsewhere. The blended surface, as depicted in Fig. 1 $(right)$, is smooth over the transitions and switch to an IBS patch in the interior. The proposed method provides a good trade-off between the global fitting, like RBF [14], that delivers a single analytic function and the local fitting methods, like Multi-level Partition of Unity (MPU) [15], that is based on small local quadratic patches.

The main features of our approach are summarized as follows: (i) a sparse system of equations is presented to find the implicit surface through an algebraic fitting that incorporate local geometric information; (ii) this technique is further accelerated to reduce the dimension to the active parameters around the object; (iii) further control on the surface are allowed through the smoothness parameter and the user interaction; (iv) a novel weighting technique has been

introduced in order to blend small patches together in the overlapping regions and leave them unchanged in the interior.

In the next section related works are briefly reviewed and classified based on the solution space and their fitting method. Section 3 provides the preliminaries for the problem formulation. In Section 4 the 3L-IBS algorithm for local patch reconstruction is presented. Then, we show how the IBS surface can be further controlled by a quadratic regularization term and additional constraints. Finally, a novel weighting technique is introduced to blend the reconstructed IBS patches. The experimental results and comparisons with state of the art are presented in Section 5 demonstrating the advantage of the proposed algorithm. Finally, conclusions are given in Section 6.

II. Related Work

The most relevant works on surface reconstructions are reviewed and classified based on the solution space and the fitting methodology. The solution space determines the domain where the shape parameters are defined. Moving on this space will change the shape of zero set, and the objective of fitting is to find the best parameters. The goodness of fitting is measured through the distance function which is explained in the second subsection.

A. Solution Space

The objective of implicit fitting, in general, is to describe a given set of points through the *zero set* of an implicit function $f_{\mathbf{c}}$: $Z_{f_{\mathbf{c}}} = \{\mathbf{x} : f_{\mathbf{c}}(\mathbf{x}) = 0\}$. The fitting procedure seeks for the best coefficient vector $\mathbf{c}$ that minimizes the distance between the given points and the zero set. Implicit Polynomial (IP) provides one of the simplest solution spaces for surface reconstruction. An IP of degree d is defined as:

$$f_{\mathbf{c}}(\mathbf{x}) = \sum_{\substack{(i+j+k)\leq d \\ \{i,j,k\}\geq 0}} c_{i,j,k} x^i y^j z^k \tag{1}$$

where $\{c_{i,j,k}\}$ are the IP coefficients and $\{x^i y^j z^k\}$ are the monomials. This definition can be viewed as the inner product of the monomial vector $\mathbf{m}(\mathbf{x})$ and the coefficient vector $\mathbf{c}$ by considering the same order on both sets.

Implicit polynomial offers a smooth and compact representation of the object through a simple coefficient vector. IPs are defined linearly with respect to the coefficient vector, and this property makes them popular for linear fitting, as explained in the next section. Unfortunately, these monomials are not locally supported[1] and, as a consequence, any change in the coefficients can lead to global changes in the whole shape of the zero set. In addition, the linear systems of IP coefficients may suffer from the instability problem [16].

Radial Basis Function (RBF) provides a good alternative to IPs through some special basis functions. These functions are called radial since their values in each point only depend on the distance from the control points [17]. Linear combinations of these basis functions are widely used for surface approximation:

$$f(\mathbf{x}) = P(\mathbf{x}) + \sum_{i=1}^{m} c_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) \tag{2}$$

where P is a low degree IP, $\{\mathbf{x}_i\}_1^m$ are the control points and $\mathbf{c}$ is the coefficient vector. Common choices for the radial function includes $\phi(r) = r^k$ and $\phi(r) = r^2 \log r$. The authors in [14] and [18] use RBFs to reconstruct the surface from 3D range data. They present a regularization method for RBF that helps to describe the object in different smoothness orders.

Implicit B-Spline (IBS) offers a flexible representation using some specific basis functions that are locally supported and guarantee the smoothness [19]. IBS is defined through the linear combinations of B-Splines tensor products:

$$f(\mathbf{x}) = \sum_{i,j,k=1}^{N} c_{i,j,k} B_i(x) B_j(y) B_k(z) \tag{3}$$

where $B_i(x)$, $B_j(y)$ and $B_k(z)$ are the spline basis functions, $\{c_{i,j,k}\}$ are the coefficients defining a control lattice of size $N \times N \times N$. Fig. 2$(right)$ depicts how an IBS provides a more flexible representation than an IP. Moreover, since it is linear with respect to its coefficient vector it is appropriate for linear fitting [20] as well as the non-linear ones [21], [22]. In Section IV, we show how these functions can be employed in a fast linear fitting accompanied by the regularization energy and additional constraints.

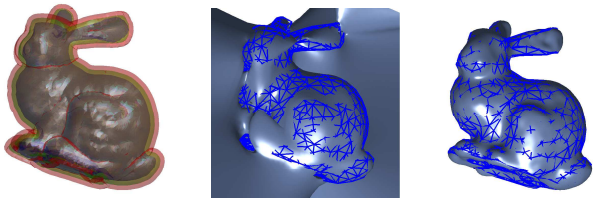

Fig. 2. Fitting the *Bunny* data set containing 817 points; $(left)$ 3L offsets; $(middle)$ reconstructed implicit polynomials [23]; $(right)$ reconstructed implicit b-spline.

B. Fitting Methodology

Having confirmed the solution space, the fitting procedure seeks the best parameter in this space using a proper fitting metric. This distance function measures the closeness of the given data points to the zero level surface. Geometric approaches approximate the orthogonal distance and try to minimize it through some gradient-based method, while the Algebraic approaches are based on some simple rough estimation of the distance. In this subsection we review the most relevant fitting techniques used by computer vision [2], CAD [20] and graphics communities [1].

Let assume that $\mathcal{P} = \{\mathbf{p}_i\}_1^n$ is the set of given points sampled from the object surface. A good fitting function is supposed to reach zero close to these points; hence, as a simple criterion, the deviation of function values from zero can be

[1] support of an implicit function is where it doesn't vanish.

used for measuring the distance between the points and the zero set:

$$E = \sum_{i=1}^{n} f^2(\mathbf{p}_i) \tag{4}$$

this accumulated residual error can be viewed as a function of the coefficient vector $\mathbf{c}$, which is linear for the cases of IP, RBF and IBS functions: $f(\mathbf{p}_i) = \mathbf{m}(\mathbf{p}_i)^{\mathrm{T}}\mathbf{c}$. Therefore, the algebraic distance can be defined as a quadratic term:

$$E = \|\mathbf{M}_{\mathcal{P}}\mathbf{c}\|^2 = \mathbf{c}^{\mathrm{T}}\mathbf{M}_{\mathcal{P}}^{\mathrm{T}}\mathbf{M}_{\mathcal{P}}\mathbf{c} \tag{5}$$

where the monomial matrix $\mathbf{M}_{\mathcal{P}}$ is constructed of the monomial vectors $\mathbf{m}(\mathbf{p}_i)^{\mathrm{T}}$ in a row-by-row manner. Minimization of this quadratic term is widely known as Least Squares (LS). In order to avoid the trivial solution it must be accompanied with some constraints on the coefficient vector.

The least squares solution can be computed quite fast by solving a system of linear equations, which is usually unstable. A better fitting metric has been proposed in [24] by using the first order distance approximation. Indeed, it considers the absolute values normalized by the gradient $|f|/\|\nabla f\|$ instead of f in (4). The 3L algorithm is another fitting technique that exploits more geometric clues of the point set [12]. In this method the original data is supported by two additional offsets in order to control the function values around the zero set. The 3L algorithm has been originally proposed for implicit polynomial fitting as detailed in the next section.

The associated normal vectors can be also incorporated during the surface fitting. Poisson surface reconstruction finds the best *indicator function* whose gradient vectors are compatible with the normal vectors by satisfying a Poisson equation. Similarly, in [25] the associated normal vectors are considered to find the best IP. The optimal function is found such that its gradient vectors in the data set are parallel to the associated normal vectors. This method is referred to as the Gradient-one in the literature, since it considers the normals with the unit length. The 3L algorithm and the Gradient-one have been extended in [16] and [26] to improve the instability problem using a ridge regression method. In this work the instability problem is tackled by using a thin-plate energy term that is geometrically meaningful.

The above-mentioned methods are referred to as *global fitting*, where a single analytic function is obtained to describe the whole shape. On the contrary there are many *local fitting* methods, where the function value in each point is computed based on the local information around that point. Moving least squares (MLS), for instance, is based on a weighted least squares defined in a local neighborhood [27]. Multi-level Partition of Unity is another local fitting method that is widely used for surface reconstruction [15], [28], [29]. Generally speaking, MPU describes the object based on a set of small patches that blend together in a smooth way. The whole set of points is recursively structured as an oct-tree, where each cell is described by a low degree IP (e.g., quadric). Finally, all these local approximations are smoothly integrated into a global function. In [28] the authors use the gradient-one algorithm and ridge regression techniques to obtain a better result for a local description. In this work we show how the proposed IBS fitting equipped with an efficient weighting function provides a good trade-off between global and local fitting methods.

III. 3L-IP Fitting

The 3L algorithm, as mentioned in section II-B, is an algebraic IP fitting technique that exploits local geometric clues in the data set [12]. It generates an inner and outer offset denoted by $\mathcal{P}_{-\delta}$ and $\mathcal{P}_{+\delta}$ at the distance $\pm\delta$ from the original data set $\mathcal{P}_0$. These sets are generated along the normal vectors that can be obtained either through the triangulation [16] or through the PCA applied in each local neighborhood. Fig. 2$(left)$ depicts how the additional offsets support a data set in 3D.

Having provided all the three level sets: $\{\mathcal{P}_{-\delta}, \mathcal{P}_0, \mathcal{P}_{+\delta}\}$, the 3L algorithm finds an implicit polynomial approaching $+\epsilon$ inside, $-\epsilon$ outside and zero in the original data set. This set of constraints can be formulated as an over-determined system of equations $\mathbf{M}_{3L}\mathbf{c} = \mathbf{b}$ where

$$\mathbf{M}_{3L} = \begin{bmatrix} \mathbf{M}_{\mathcal{P}_{-\delta}} \\ \mathbf{M}_{\mathcal{P}_0} \\ \mathbf{M}_{\mathcal{P}_{+\delta}} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} +\boldsymbol{\epsilon} \\ \mathbf{0} \\ -\boldsymbol{\epsilon} \end{bmatrix} \tag{6}$$

$\mathbf{M}_{3L}$ is a block matrix of $\mathbf{M}_{\mathcal{P}_{+\delta}}$, $\mathbf{M}_{\mathcal{P}_0}$ and $\mathbf{M}_{\mathcal{P}_{-\delta}}$ containing the monomial vectors calculated in the point set. The column vector $\boldsymbol{\epsilon}$ contains the expected value of f in the additional offsets. Since the zero set is the same for $\mathbf{c}$ and $\kappa\mathbf{c}$ for any constant κ, we can fix the value $\epsilon = 1$ without loss of generality.

Solving the overdetermined system of the 3L algorithm in (6) is equivalent to minimizing the following *data term*:

$$\begin{aligned} E(\mathbf{c}) &= \|\mathbf{M}_{3L}\mathbf{c} - \mathbf{b}\|^2 \\ &= \mathbf{c}^{\top}\mathbf{M}_{3L}^{\top}\mathbf{M}_{3L}\mathbf{c} - 2\mathbf{c}^{\top}\mathbf{M}_{3L}^{\top}\mathbf{b} + \mathbf{b}^{\top}\mathbf{b} \end{aligned} \tag{7}$$

It should be highlighted that this least squares problem does not lead to the trivial solution for low degree IPs; hence, no extra constraint is required. This function is quadratic with respect to the coefficient vector and its derivatives can be easily computed using matrix calculus:

$$\partial E/\partial \mathbf{c} = 2(\mathbf{M}_{3L}^{\top}\mathbf{M}_{3L})\mathbf{c} - 2\mathbf{M}_{3L}^{\top}\mathbf{b} = \mathbf{0} \tag{8}$$

Then, the least squares solution can be computed by solving this system of linear equations. This algorithm is quite fast and more stable than the standard algebraic fitting. It finds the solution directly from the monomial matrix without any iterative computation.

In spite of being computationally efficient, the 3L algorithm suffers from some problems. The first one is due to the numerical instability that is common in linear least squares. In [16] a ridge regression (RR) method is presented to increase the stability by adding a diagonal matrix, which can be chosen as the identity matrix. The authors in [16] impose some conditions on this matrix in order to obtain an algebraic-invariant result. Although more stable, the RR method decreases the accuracy. The second problem is regarding the limitation of implicit polynomials for defining the solution space. Increasing the IP degree allows us to describe more complex objects, but it creates some artifacts and outliers around the object.

IV. Proposed Method

In this section, the proposed method for implicit B-spline surface reconstruction is explained. First of all, the 3L algorithm is adapted for IBS fitting in order to provide a more flexible shape description. We show how the surface reconstruction can be controlled by considering a proper regularization term and imposing some additional constraints. Although the 3L-IBS surface reconstruction is solely strong enough for shape description we show how this tool can be further improved through a blending technique. The resulting method is very fast and it provides a good trade-off between the local and global fitting methods. All the formulations in this article are easy to implement since a simple definition of IBS has been used.

A. 3L-IBS Patch Reconstruction

Implicit B-splines provide a very flexible representation that allows the local control and avoids artifacts (Fig. 2). In this section, we consider this tool to define the solution space of the 3L algorithm. The resulting method, referred to as 3L-IBS, is much faster than the original 3L-IP since it leads to a sparse system of equations due to the compact supports. For the sake of simplicity, all the formulations are presented based on the *blending functions*, which are small quadratic patches. As illustrated in Fig. 3 these small patches are translated and blended together in order to construct the B-Spline basis functions that is $\mathcal{C}^2$-continuous.

As a key point, the IBS definition in (3) can be viewed as the inner product of the coefficient vector $\mathbf{c}$ that is to be found and the basis vector $\mathbf{m}(\mathbf{x})$ that depends on the given point:

$$f(\mathbf{x}) = \mathbf{c}^\top \mathbf{m}(\mathbf{x}) = \mathbf{m}(\mathbf{x})^\top \mathbf{c} \tag{9}$$

where the same order has been used for vectorizing the control values $\{c_{i,j,k}\}$ and the basis functions $\{B_i(x)B_j(y)B_k(z)\}$. In the case of cubic splines with uniform *knot*[2] sequence, the b-spline functions can be simply constructed through the blending patches:

$$\begin{aligned} b_0(u) &= (1-u)^3/6, \quad b_1(u) = (3u^3 - 6u^2 + 4)/6, \\ b_2(u) &= (-3u^3 + 3u^2 + 3u + 1)/6, \quad b_3(u) = u^3/6 \end{aligned} \tag{10}$$

Fig. 3 shows how these blending functions are rescaled and translated to construct the B-splines in 1D. Using these cubic blending functions leads to a C^2 continuity.

Without loss of generality we normalize the point cloud into a unit cube. For defining N basis functions over the unit interval $[0, 1]$ the step of the knot sequence must be chosen as $\Delta = 1/(N-3)$ (see Fig. 3). Then, the IBS definition in (3) can be directly presented based on the blending functions in (10):

$$f(\mathbf{x}) = \sum_{r,s,t=0}^{3} c_{i+r,j+s,k+t} b_r(u).b_s(v)b_t(w) \tag{11}$$

where the starting indices depend on the point coordinates as it follows:

$$i = \lceil x/\Delta \rceil, \quad j = \lceil y/\Delta \rceil, \quad k = \lceil z/\Delta \rceil \tag{12}$$

[2]knot: position where two blending functions join.

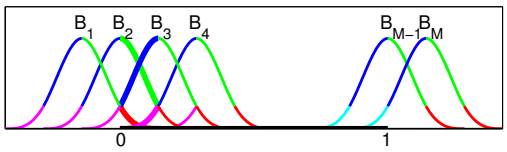

Fig. 3. Cubic b-spline basis functions are made out of blending patches that are re-scaled and translated.

and for every given point, 4^3 control coefficients are incorporated in IBS computation. Based on our setting, the original xyz-coordinates must be mapped into uvw before applying the blending functions:

$$u = \{x/\Delta\}, \quad v = \{y/\Delta\}, \quad w = \{z/\Delta\} \tag{13}$$

where $\{\alpha\} = \alpha - \lfloor \alpha \rfloor$ is the fractional part.

This simple definition of IBS should be used for computing the basis vectors in (9). Having computed the basis vectors in the given points and the additional offsets, the 3L basis matrix $\mathbf{M}_{3L}$ can be constructed row-by-row in order to find the least squares coefficient vector:

$$\mathbf{c} = \mathbf{M}_{3L}^\dagger \mathbf{b} = (\mathbf{M}_{3L}^\top \mathbf{M}_{3L})^{-1} \mathbf{M}_{3L}^\top \mathbf{b} \tag{14}$$

where $\mathbf{M}_{3L}^\dagger$ denotes the pseudo-inverse of the basis matrix. The main advantage of the 3L-IBS algorithm is that the basis matrix $\mathbf{M}_{3L}$ is very sparse for the case of implicit B-splines (4^3 non-zero elements in each row), while for the case of IP the basis matrix is highly dense. This fact makes the 3L-IBS algorithm much faster than the original 3L algorithm, while it benefits from a more flexible IBS representation.

B. Regularization

The basis matrix in (14) is very sparse and it could lead to more than one optimal solution. Indeed, some of the control coefficients do not have any contribution during the least squares. As explained in Section III, this problem can be tackled by ridge regression by adding a proper diagonal matrix, but it is not geometrically meaningful. In the current work a global tension term is considered for regularization, which estimates the *curvature* of f over the whole domain Ω [20].:

$$T(\mathbf{c}) = \iiint_\Omega f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2 + 2f_{xz}^2 + 2f_{yz}^2 + f_{zz}^2 dxdydz \tag{15}$$

This term is added to the objective function of 3L-IBS by using a proper regularization parameter μ:

$$E(\mathbf{c}) = \|\mathbf{M}_{3L}\mathbf{c} - \mathbf{b}\|^2 + \mu T(\mathbf{c}) \tag{16}$$

This is similar to the bending energy used in Thin Plate Spline (TPS) for spatial deformations. Fig. 4 illustrates the effect of regularization parameter on the rigidity of the IBS surface. In general, a high regularization parameter results in a more rigid surface, while a lower value leads to a more relaxed one.

The regularization term is a quadratic function of the coefficient vector that can be analytically pre-computed for the given size of IBS. As detailed in Appendix A, the regularization term can be represented through the matrix notation:

$$T(\mathbf{c}) = \mathbf{c}^\top \mathbf{H} \mathbf{c} \tag{17}$$

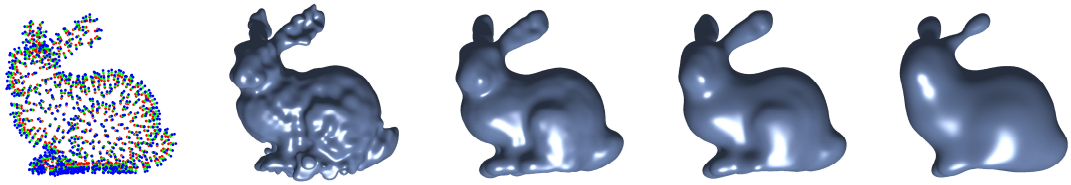

Fig. 4. The effect of regularization parameter in 3L-IBS method; $(left)$ the offset of *Bunny* ($\delta = 1\%$); the rest from left-to-right: different settings for the regularization parameter $\mu = 1$, 10^2, 10^3, 10^4 respectively.

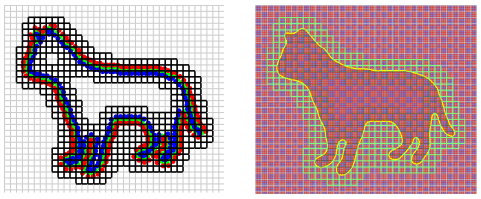

Fig. 5. The truncated IBS curve: $(left)$ the active control points are highlighted; $(right)$ the efficient algorithm considers only a narrow stripe around the active control points.

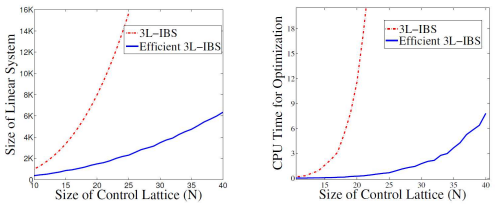

Fig. 6. Complexities of the 3L-IBS and the truncated version for *Bunny* data set containing $10k$ points: $(left)$ the size of the optimization problems; $(right)$ the CPU-time spent for finding the optimal control lattice.

where $\mathbf{H}$ is a $N^3 \times N^3$ sparse matrix including the integral of overlapping basis functions and their derivatives. Having this matrix pre-computed, the optimal IBS coefficient vector can be easily found by solving a sparse system of equations:

$$\mathbf{c} = (\mathbf{M}_{3L}^\top \mathbf{M}_{3L} + \mu \mathbf{H})^{-1} \mathbf{M}_{3L}^\top \mathbf{b} \tag{18}$$

It should be highlighted that only a few number of coefficients may contribute to construct the IBS zeros set. Fig. 5 illustrates the active coefficients associated to the zero set in a 2D case. Knowing this fact, we can easily come up with a very efficient version, referred to as *Truncated IBS*, that only finds the active coefficients. Fig. 6 shows how this efficient version saves a lot of computational cost.

C. Additional constraints

In addition to the data term used for finding the optimal parameters, the user can interactively impose some constraints on the reconstruction. These constraints force the zero set to pass through some specific points or to obtain a specific orientation there; they are referred to as positional and orientation constraints, respectively. We show that the proposed framework can be upgraded to allow these interactions.

Lets consider the points $\mathbf{q}_1, ..., \mathbf{q}_k$ as the positional constraints provided by the user. In order to force the IBS zero set pass through these point, we must impose the constraints of the form $f(\mathbf{q}_i) = 0$ on the optimization problem in (16). Each constraints is linear with respect to the coefficient vector as $\mathbf{w}_i^\top \mathbf{c} = 0$, where $\mathbf{w}_i = \mathbf{m}(\mathbf{q}_i)$ is the basis vector calculated in the constraint point. Considering all these constraints, a constraint matrix $\mathbf{W}$ is constructed that contains all these basis vectors. The constrained least squares is formulated as:

$$\begin{aligned} &\min_{\mathbf{c}} E(\mathbf{c}) = \|\mathbf{M}_{3L}\mathbf{c} - \mathbf{b}\|^2 + \mu \mathbf{c}^\top \mathbf{H}\mathbf{c} \\ &s.t. \\ &\quad \mathbf{W}\mathbf{c} = \mathbf{0}. \end{aligned} \tag{19}$$

In order to solve this problem, the Lagrange multiplier method can be used [12], where the Lagrange function is defined as:

$$\Lambda(\mathbf{c}, \lambda) \quad = \quad E(\mathbf{c}) - \sum_{i=1}^{k} \lambda_i \mathbf{w}_i^\top \mathbf{c} \tag{20}$$

Since all the terms are either linear or quadratic, it is easy to compute the derivatives through matrix calculus. The derivative with respect to λ obviously leads to the positional constraints $\mathbf{W}\mathbf{c} = \mathbf{0}$, while the ones with respect to $\mathbf{c}$ result in:

$$2(\mathbf{M}_{3L}^\top \mathbf{M}_{3L})\mathbf{c} - 2\mathbf{M}_{3L}^\top \mathbf{b} + 2\mu \mathbf{H}\mathbf{c} = \mathbf{W}^\top \boldsymbol{\lambda} \tag{21}$$

Consequently, the optimal control vector $\mathbf{c}$ for the constrained 3L-IBS method can be easily computed as:

$$\mathbf{c} = \mathbf{A}(\mathbf{M}_{3L}^\top \mathbf{b} + 1/2 \mathbf{W}^\top \boldsymbol{\lambda}) \tag{22}$$

where $\mathbf{A} = (\mathbf{M}_{3L}^\top \mathbf{M}_{3L} + \mu \mathbf{H})^{-1}$ and the unknown Lagrange multiplier $\boldsymbol{\lambda}$ can be easily found by multiplying (22) by $\mathbf{W}$:

$$\boldsymbol{\lambda} = -2(\mathbf{W}\mathbf{A}\mathbf{W}^\top)^{-1} \mathbf{W}\mathbf{A}\mathbf{M}_{3L}^\top \mathbf{b} \tag{23}$$

Therefore, the proposed framework for surface reconstruction allows regularization and user interaction at the same time just by solving a system of linear equations. Fig. 7 shows a case where the fitting curve does not interpolate all the points. After introducing three positional constraint by the user, the constrained 3L-IBS in (22) perfectly interpolates these points. In a similar way, orientation constraints can be introduced in the formulation by defining the right constraint matrix $\mathbf{W}$ in (19); the rest remains unchanged since these constraints are still in the linear form.

D. Blended 3L-IBS Algorithm

The 3L-IBS algorithm proposed in this manuscript is very fast and flexible since it finds the best IBS coefficients through solving a sparse system of equations. This fact makes it very attractive for surface reconstruction from large clouds of

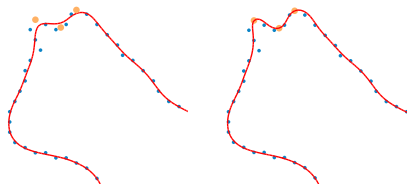

Fig. 7. The user interaction is allowed in the proposed framework. $(left)$ three constraints are added by the user; $(right)$ the constrained IBS passes through these constraints.

Fig. 8. Blended IBS: $(left)$ different patch sizes ($N = 10, 20$); $(right)$ the blended surface using the proposed weighting technique.

points. Moreover, it is shown how the final result can be regularized and further controlled through the user interactions. In general, the proposed algorithm provides a global fitting, where a single analytic function describes the whole object through its zero set. In this section we show how the power of 3L-IBS can be extremely enhanced by blending some mid-size IBS patches. The resulted method provides a good trade-off between the global and local fitting approaches.

Partition of unity approach is a technique to blend some local patches in order to construct a global function that is smooth over the transition regions. As the first step, the 3L-IBS algorithm is able to reconstruct mid-size IBS patches with a very low computational cost. Each patch describes a partition of the object and may overlap another patch in the neighborhood. Partition of unity helps one to construct a global function out of small IBS patches. Fig. 8 illustrates two different sizes of IBS patches for $N = 10, 20$ respectively. In the first one, some lower size IBS patches are reconstructed each by solving a sparse system of equations, while in the second one, some highly detailed IBS patches are reconstructed. We show how these patches can be smoothly blended in the overlapping regions as illustrated in the right column.

First, a typical partition of unity approach is presented and then we explain how a new weighting function is designed in order to partially blend the patches. Lets consider the local IBS patches as $f_1(\mathbf{x}), ..., f_n(\mathbf{x})$. The objective is to blend these patches to build a high resolution global implicit function $F(\mathbf{x})$. Partition of unity makes it possible through a set of weighing functions $\hat{w_1}(\mathbf{x}), ..., \hat{w_n}(\mathbf{x})$ that sum up to one over the domain Ω:

$$\sum_{i=1}^{n} \hat{w_i}(\mathbf{x}) = 1, \forall \mathbf{x} \in \Omega \tag{24}$$

Indeed, these weighting functions provide a partition of unity over Ω. Moreover, they should be locally supported in order to only include the neighboring patches in the given point. Afterwards, the global fitting function F can be simply constructed from the local patches:

$$F(\mathbf{x}) = \sum_{i=1}^{n} \hat{w_i}(\mathbf{x}) f_i(\mathbf{x}) \tag{25}$$

In other words, the global function F is a convex combination of the local patches, whose contributions depend on the weighting functions $\{\hat{w_i}\}_1^n$.

In this work we present a novel weighting technique to blend the local IBS patches. In fact, we aim to blend the patches just in the overlapping regions and let the IBS to remain the same in the interior regions. Fig. 9$(left)$ illustrates a typical cell of size SC in a regular grid and its narrow regions of width d where the IBS patches overlap. The interior region C_i^0 is where the local patch remains unchanged. In [15] the weighting function is designed such that it increases closing to the center of the cell $\mathbf{c}_i$. In our work, the weighting function is based on the distance of the given point to the interior region:

$$dist(x, C_i^0) = \Big[\|\mathbf{x} - \mathbf{c}_j\|_p - (\frac{SC}{2} - d)\Big]_+ \tag{26}$$

where, $\|.\|_p$ is the L^p norm and $[t]_+ = max(t, 0)$. Then, the distance value is zero for the whole interior region and it increases by getting farther from the boundary. Afterwards, the weighting function can be defined as follows:

$$w_j(\mathbf{x}) = b\Big(\alpha.dist(x, C_j^0)\Big) \tag{27}$$

where $b(.)$ is the quadratic B-Spline basis function with the support in $[0, 1.5]$ and $\alpha = 1.5/d$ is used to re-scale the interval $[0, d]$ to the support interval. Fig. 9$(right)$ shows how the weighting function vanishes by getting far from the interior region, while it remains constant in that region. In order to satisfy the condition in (24) for being a partition of unity, the normalization must be applied on the weighting functions:

$$\hat{w_i}(\mathbf{x}) = \frac{w_i(\mathbf{x})}{\sum_{j \in \mathcal{N}_i} w_j(\mathbf{x})} \tag{28}$$

where $\mathcal{N}_i$ is the neighboring cells of C_i.

Having defined the weighting functions in (27) and normalized them in (28) we can handle the patch blending to come up with the global function F defined in (25). Fig. 8$(right)$ illustrates an example of IBS patch blending through the proposed weighting functions. The transitions between the patches are very smooth since the weights in (27) are differentiable. Moreover, the IBS patches do not change in

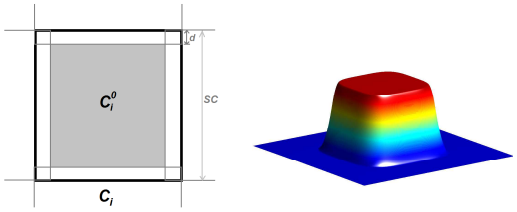

Fig. 9. Blending components in 2D: $(left)$ The overlapping cells used to subdivide the domain; C_i^0 shows the interior region; $(right)$ the weighting function defined in (27) for blending IBS patches in 2D.

the interior region since $\hat{w}_i(\mathbf{x}) \equiv 1$ over this region; hence, the blending happens only in the overlapping regions. As a consequence, our proposed *blended IBS surface reconstruction* technique provides a good trade-off between the global fitting (e.g., IBS) and local fitting techniques (e.g., MPU or MLS).

V. Experimental Results

Experimental results together with comparisons with the state of the art fitting techniques are presented in this section. The comparisons are applied for both 2D and 3D data sets, which are either public [13], [30] or obtained through a 3D scanner (i.e., Konica Minolta Vivid 9i) or multi-camera platform [31]. The comparisons are based on the geometric accuracy as well as the CPU time and the reconstruction time.

The fitting accuracy of both IP and IBS is measured through the accumulated fitting error (AFE). It measures the geometric distances between the data points and the zeros set of the fitting result. Hence for every single data point $\mathbf{p}$ its corresponding foot-point $\tilde{\mathbf{p}}$ must be iteratively estimated. This point lies on the surface and the line between $\mathbf{p}$ and $\tilde{\mathbf{p}}$ must be parallel to the gradient vector at the foot-point:

$$\begin{pmatrix} f(\tilde{\mathbf{p}}) \\ \nabla f(\tilde{\mathbf{p}}) \times (\tilde{\mathbf{p}} - \mathbf{p}) \end{pmatrix} = \mathbf{0} \tag{29}$$

This equation can be iteratively solved via Newton method for non-linear system of equations. In each iteration, the point moves to a lower level while its gradient direction gets parallel to the connecting line.

A. Quantitative Comparisons

Several point clouds in 2D and 3D have been reconstructed with the proposed approach and quantitatively compared with the best results obtained with the original 3L algorithm [12], and the Gradient-one [25], [20]. Fig. 10 illustrates six data sets with different complexities. In the second column the zero sets of the optimal IPs obtained by the original 3L algorithm are presented (the IP coefficient sizes are provided in Table I). The third and forth columns correspond to the optimal IBS obtained by the Gradient-one and the 3L-IBS algorithm. The results obtained though IPs ($2nd$ column) clearly suffers from outliers and inaccuracies. This problems gets worse with increasing the object's complexity.

The main problem of implicit polynomials is the unbounded supports of their monomials. In other words, none of the monomials vanish in infinity and it makes all the coefficients

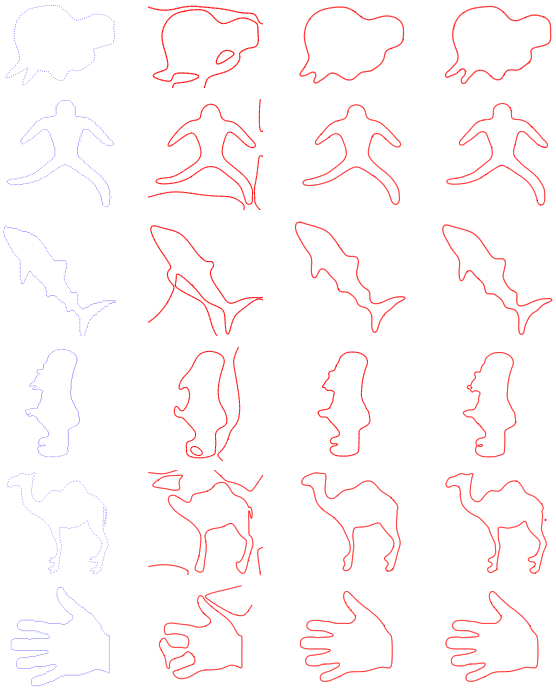

Fig. 10. (1^{st} col.) Original 2D data sets. Results obtained, for the zero-sets, by: (2^{nd} col.) 3L-IP [23]; (3^{rd} col.) Gradient-one algorithm; (4^{th} col.) the proposed 3L-IBS.

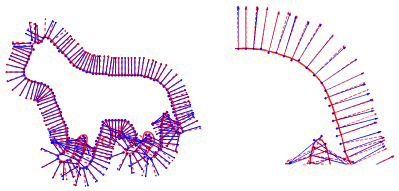

Fig. 11. $(left)$ The result obtained by the Gradient-one algorithm [25]; $(right)$ the enlargement: blue vectors for the estimated normals and red dashed-lines for the curve normals. It shows how this fitting algorithm ignores the position information.

contribute in the curve/surface reconstruction for any given point. On the contrary, the cubic IBS curves have compact supports while they meet $\mathcal{C}^2$ continuity. As illustrated in the $3rd$ and $4th$ columns of Fig. 10 the optimal IBS curves are smooth with no artifacts. Moreover, as stated in Table I, the optimal IBSs achieve lower fitting error than IPs and the Gradient-one method [20]; the latter has a similar accuracy to our method, but since our method is more based on the point set positions than their orientation, the final zero set is closer to the data set (see Fig. 11).

In addition to high accuracy, the proposed method is much faster in both optimization and reconstruction. As stated in Table I the size of the parameter space in the case of Truncated IBS is much smaller than the Gradient-one. Actually, as already depicted in Fig. 6 the parameter size increases quadratically with respect to the control lattice resolution N, while in the Gradient-one case, it increases in a cubic growth. Hence in our case the least squares is applied for a sparse system with smaller number of parameters. This point is even more highlighted when it comes to the 3D surface

TABLE I
SIZE OF SOLUTION SPACES AND ACCUMULATED FITTING ERRORS (AFES) FOR COMPARISON BETWEEN DIFFERENT APPROACHES ON **2D DATA SETS** (3L: 3L ALGORITHM; GO: GRADIENT ONE; PA: PROPOSED APPROACH).

	3L[12]		**GO**[25]		**PA**	
Figure	Size	AFE	Size	AFE	Size	AFE
Fig. 10-(oni)	231	1.0148	400	0.9957	190	0.2788
Fig. 10-(dude)	351	1.0612	625	1.3362	305	0.6183
Fig. 10-(fish)	231	2.1222	400	0.8979	178	0.7475
Fig. 10-(homer)	496	1.2625	900	0.5797	303	0.2546
Fig. 10-(camel)	861	2.6782	1600	1.1398	603	0.7668
Fig. 10-(hand)	1081	6.2209	2500	0.8444	865	0.4000

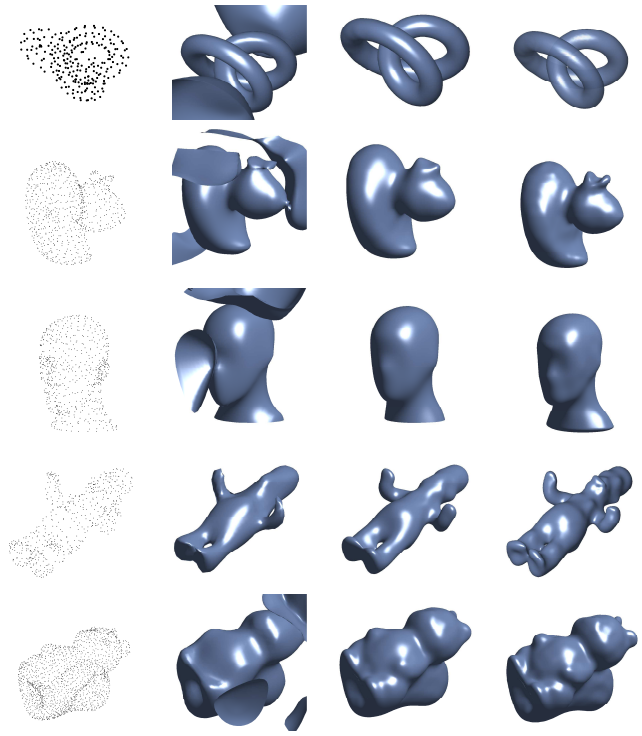

Fig. 12. Surface reconstruction for low densities: (1^{st} col.) original 3D data sets. Results obtained, for the zero-sets, by: (2^{nd} col.) 3L-IP [23]; (3^{rd} col.) Gradient-one algorithm; (4^{th} col.) the proposed 3L-IBS.

reconstruction.

Surface reconstruction from low density point clouds have been studied in Fig. 12 and Table II. For all of these examples low degree IPs ($d = 10 \sim 20$) and low resolution IBS ($N = 20 \sim 30$) are used. As illustrated in the second column, IP reconstruction usually leads to many artifacts around the zero level, while IBSs deliver more stable results in a lower time. In fact the IP coefficients corresponding to the high degree monomials can disturb the final result. In the IBS cases, the degree of the basis functions do not exceed the cubic degree. The fitting accuracy and CPU times spent for least squares optimization, as presented in Table II, highlight the relative accuracy and efficiency of the presented framework.

B. Qualitative Comparisons

In this part of experiments, high density point clouds are considered for qualitative comparisons. Fig. 13 shows the

TABLE III
HIGH RESOLUTION 3D DATA SETS; (DAT: NUMBER OF DATA POINTS; LAT: SIZE OF CONTROL LATTICE; ACT: SIZE OF ACTIVE CONTROL LATTICE;CPU: TIME (IN SECONDS) FOR SOLVING THE LEAST SQUARES).

	Name	**DAT**	**LAT**	**ACT**	**CPU**
Fig. 13-($1st$)	*Kitten*	11039	30	2799	0.44
Fig. 13-($2nd$)	*Hand*	16519	40	3090	0.46
Fig. 13-($3rd$)	*Bimba*	8857	40	5721	1.07
Fig. 13-($4th$)	*Armadillo*	8260	50	7728	1.27
Fig. 13-($5th$)	*Cow*	10850	50	4699	0.71

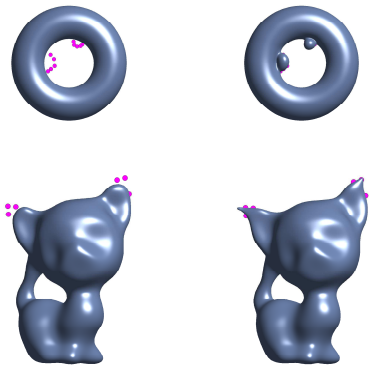

Fig. 14. User interaction on the reconstructed surface: the IBS surfaces are shown before and after imposing the user positional constraints.

surfaces reconstructed through the truncated IBS for some standard data sets [13]. In these examples, the details in the point sets are perfectly preserved by the optimized zero level. Table III presents the details about the point clouds as well as the surface parameters. In average less than $1sec.$ is spent for solving the sparse system of equations since the truncated IBS drastically reduces the computations.

Implicit surfaces and level set representations, in general, are known to be very efficient for 3D surface editing, including the primitive bolean operations as well as cut-paste editing [32]. As discussed in Section IV-C our proposed approach allows user editing by introducing the positional and orientation constraints. Fig. 14 illustrates the effect of user interaction with the reconstructed surface. The constraints are introduced in the inner ring of *Torus* and around the ears of *Kitten* and the final surfaces perfectly pass through these points, regardless of the regularization effect for the smoothness. In fact, the final surface is still reconstructed by solving a sparse system of equations as presented in (22).

For reconstructing very complex and detailed point clouds one may increase the IBS degrees of freedom, which results in a huge system of equations. As proposed in Section IV-D, instead of reconstructing a single analytic IBS surface, mid-size IBS patches can be reconstructed and blended smoothly. Fig. 15 illustrates the accuracy of the blended IBS that is reconstructed from the *Bouncing* and *Handstand* data sets in [31]. In this figure the results of blended IBS are compared with the RBF surface reconstruction [14]. The details about the degrees of freedom, CPU time and surfacing time are presented in Table IV. For the case of truncated IBS, instead of

TABLE II
SIZE OF THE SOLUTION SPACE, CPU TIMES (IN SEC.) SPENT FOR THE OPTIMIZATION AND ACCUMULATED FITTING ERRORS (AFES) TO COMPARE THE RESULTS OBTAINED BY DIFFERENT APPROACHES ON **3D DATA SETS** (3L: THE 3L ALGORITHM; GO: THE GRADIENT ONE; PA: PROPOSED APPROACH).

	Data		**3L**[12]			**GO**[25]			**PA**		
Figure	Name	#pnt	Size	CPU	AFE	Size	CPU	AFE	Size	CPU	AFE
Fig. 2	Bunny	817	1771	1.1809	5.1848	8000	3.1369	3.8938	1461	0.2186	3.2303
Fig. 12-($1st$)	Knot	240	816	0.0783	0.3338	8000	2.6430	0.1305	1484	0.1829	0.0506
Fig. 12-($2nd$)	Duck	745	1771	0.5538	2.0325	8000	2.8977	1.6983	1511	0.2572	1.3144
Fig. 12-($3rd$)	Mannequin	639	816	0.1427	3.3689	3375	0.6973	3.9155	749	0.1021	2.7177
Fig. 12-($4th$)	Homer	585	3654	2.3692	3.1584	27000	34.8287	2.0398	1453	0.2461	1.6922
Fig. 12-($5th$)	Squirrel	1717	2925	2.5315	7.7784	27000	29.4366	3.4695	3253	0.7842	2.8557

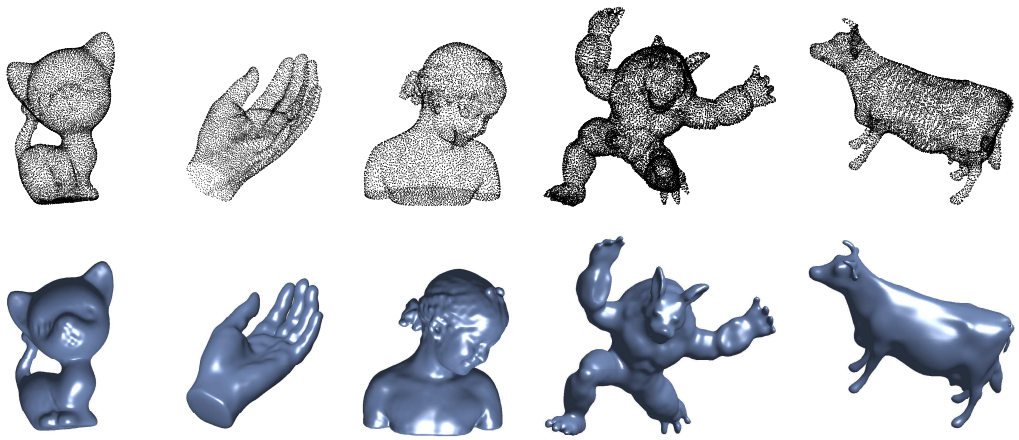

Fig. 13. (top) Original high resolution 3D data sets. ($bottom$) Results obtained by the proposed 3L-IBS.

using the regularization matrix in (15) we simply used identity matrices (Tikhonov regularization) to save time. Despite using rather lower degrees of freedom for RBF they require further computation since a full system of equations must be solved, while in our case less than one second is spent for reconstructing all the small patches. The weighting function will be incorporated when the blended surface is being reconstructed. Thanks to the use of b-spline basis functions the reconstruction time of blended IBS for a grid of $(200)^3$ voxels is around 3 seconds. Please note that how our efficient weighting system in (27) is only applied in the overlapping regions.

Our experiments shows that the proposed reconstruction technique is quite robust in noisy point cloud with missing parts. The author's face has been scanned by *Konica Minolta Vivid 9i* including around $170k$ points. The results of RBF reconstructions, truncated IBS and Blended IBS are qualitatively compared in Fig. 16 and details about the degrees of freedom and timing are presented in Table IV. All these reconstruction techniques somehow interpolate the missing parts; the fitting quality of the blended IBS is further highlighted around the nose and eyes. In the case of truncated IBS and blended IBS, the TPS regularization term in (15) has been used to obtain better resuls; this is unlike the examples of Fig. 15, where identity matrices are used for acceleration. The RBF representations in [14] and [18] select some (or all) of point clouds to define the control points of RBF; while in our case, the control points are virtually embedded in the whole space (knot sequence over the unit cube). As illustrated in Fig. 17 the MPU technique [15] and RBF reconstruction [18] fail to describe the missing part in the arm of *Swing* data set, while the blended IBS provides a pretty good result that is smooth in this part. As a part of experience it should be mentioned at the end that the point cloud size does not increase the CPU time that much, since all it cost is a bigger matrix multiplication in the main core (18) while the size of the system does not change as it depends on the degrees of freedom. So no down-sampling is required in our technique unless for saving memory.

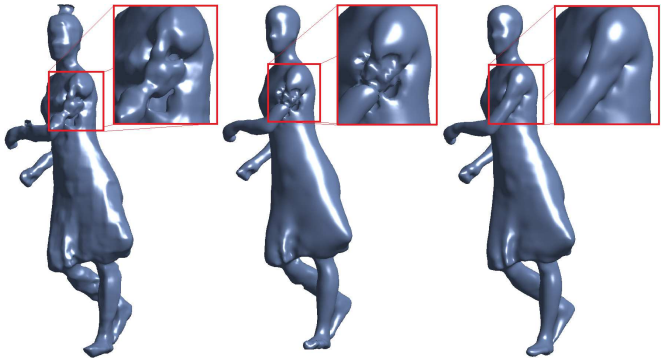

Fig. 17. Blended-IBS surface reconstruction is a good trade-off between local and global fitting: ($left$) MPU reconstruction ($\alpha = .75$); ($middle$) RBF reconstruction; ($right$) Blended IBS surface reconstruction.

VI. CONCLUSION

This paper presents a fast and flexible technique for curve/surface reconstruction from the given point cloud. The

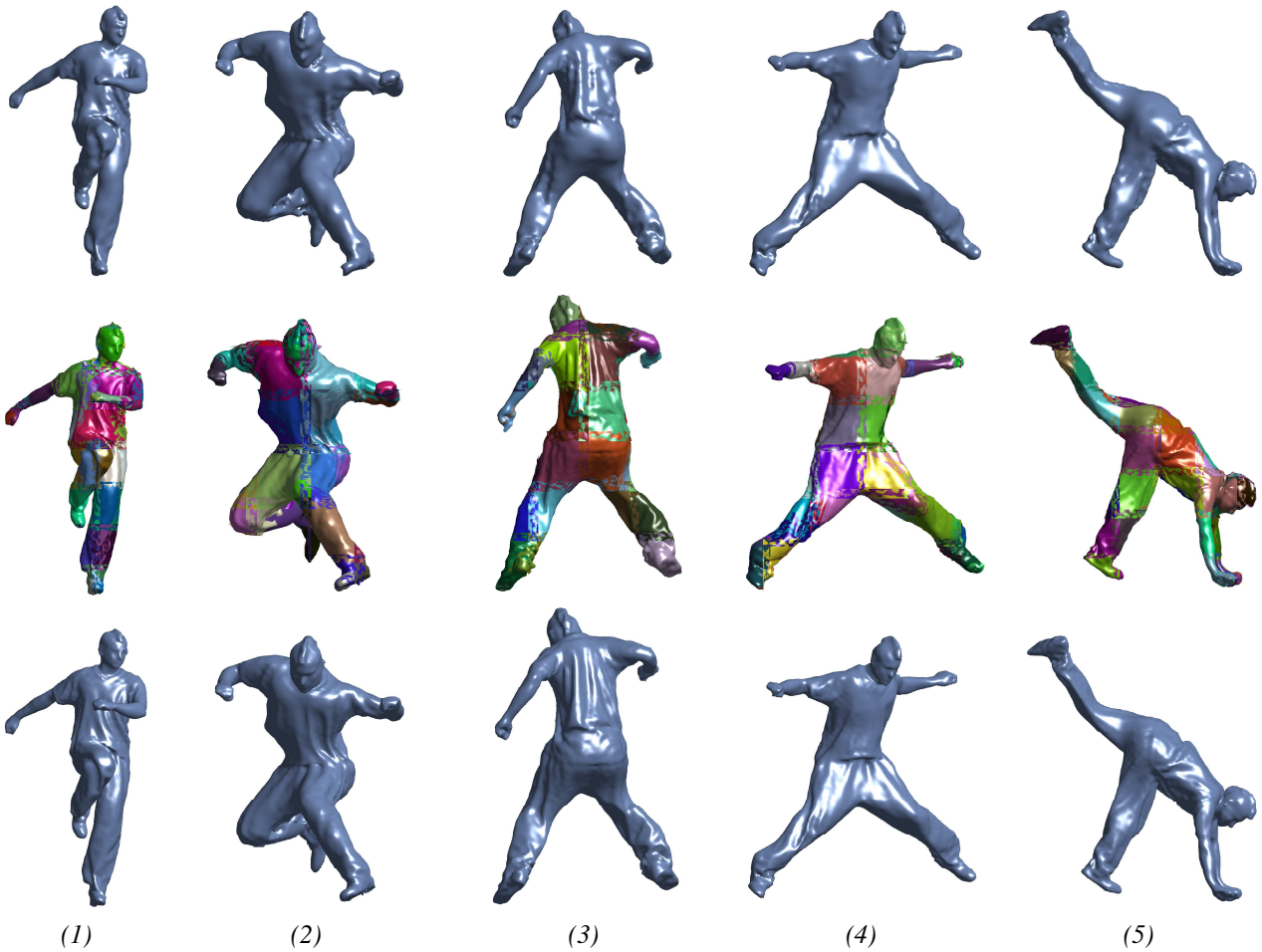

Fig. 15. Surface reconstruction of human bodies [31]: (*top*) RBF reconstruction by [14]; (*middle*) mid-size IBS patches; (*bottom*) Blended IBS surface reconstruction.

proposed approach is very fast and flexible and quite robust to noise and missing parts. The implicit b-splines have been used for the representation since they do not require any parameterization; they can be linearly described based on their parameter vector and their basis functions have compact supports. This latter fact has been exploited in this paper for proposing a fast reconstruction technique, reffered to as 3L-IBS. This technique has been further accelerated just by considering the active control parameters, which leads to a smaller sparse system of equations. The reconstructed surface can be controlled through the regularization term and the user interaction. Finally, we have proposed an efficient weighting technique for blended IBS that merge mid-size patches in the overlapping regions. The qualitative and quantitative comparisons with different reconstruction techniques show the superiority of the proposed method in time complexity and fitting quality.

APPENDIX A
REGULARIZATION TERM IN 2D

The regularization term in 2D case is a quadratic function of the control vector: $T(\mathbf{c}) = \mathbf{c}^\top \mathbf{H}\mathbf{c}$, where $\mathbf{H}$ is a $N^2 \times N^2$ symmetric matrix including the integral of basis functions' derivatives. Since the tension term T is quadratic with respect to $\mathbf{c}$ and the matrix $\mathbf{H}$ is symmetric, adding its derivative will lead to the following formula:

$$\partial E/\partial \mathbf{c} = 2(\mathbf{M}_{3L}^\top \mathbf{M}_{3L})\mathbf{c} - 2\mathbf{M}_{3L}^\top \mathbf{b} + 2\mu \mathbf{H}\mathbf{c} \qquad (30)$$

Then, setting this term to zero will give the optimal coefficient vector:

$$\mathbf{c} = (\mathbf{M}_{3L}^\top \mathbf{M}_{3L} + \mu \mathbf{H})^{-1} \mathbf{M}_{3L}^\top \mathbf{b} \qquad (31)$$

The main objective is to construct matrix $\mathbf{H}$ in (17) from the monomial vectors. For any $k, l \in \{1, 2, ..., N^2\}$, the couples (k_1, k_2) and (l_1, l_2) can be found from the table that is constructed when the active control points are detected and sorted. Hence, based on the definition, the kth entry of $\mathbf{m}_{xx}$ is equal to $(B''_{k_1}(x).B_{k_2}(y))$ (similarly for the lth entry), thus we have:

$$\begin{aligned} &\left[\iint_{XY} \mathbf{m}_{xx}\mathbf{m}_{xx}^\top dxdy\right]_{(k,l)} \\ &= \iint_{XY} (\mathbf{m}_{xx})_k.(\mathbf{m}_{xx})_l dxdy \\ &= \iint_{XY} (B''_{k_1}(x)B_{k_2}(y)).(B''_{l_1}(x)B_{l_2}(y))dxdy \\ &= \int_X B''_{k_1}(x)B''_{l_1}(x)dx. \int_Y B_{k_2}(y)B_{l_2}(y)dy \end{aligned}$$

therefore, the double integral on the whole XY domain is converted to two single integrals on X and Y domains; by applying a similar analysis on the other terms in (17) the following compact terms are obtained:

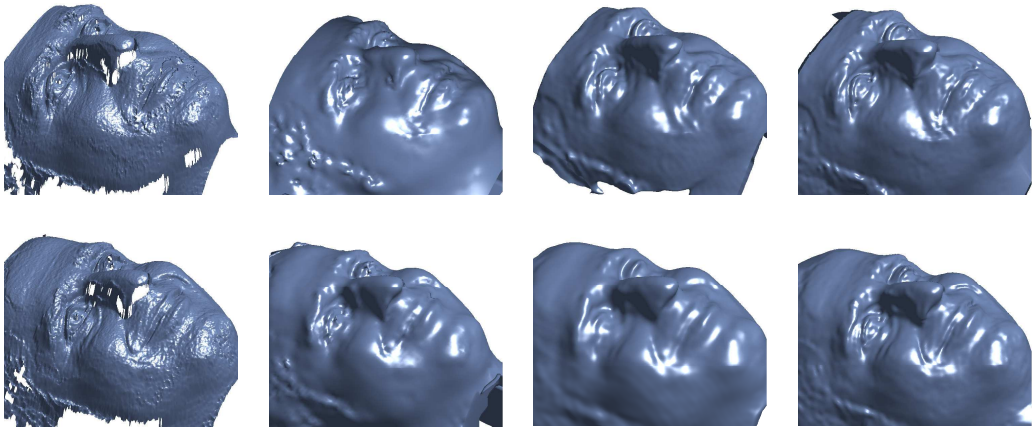

Fig. 16. 3D face reconstruction: $(1st)$ triangular meshes; $(2nd)$ RBF surface reconstruction [14]; $(3rd)$ Truncated IBS; $(4th)$ Blended IBS surface; the fitting accuracy is highlighted around the eyes and nose.

TABLE IV
COMPARISONS BETWEEN THE RBF RECONSTRUCTION [14], THE TRUNCATED IBS AND THE BLENDED IBS THAT ARE PROPOSED; DIM: DIMENSION OF THE REPRESENTATION (NUMBER OF RBF/IBS PARAMETERS); CPU: TIME (IN SECONDS) FOR SOLVING THE LEAST SQUARES; SUR: TIME SPENT FOR SURFACING.

	RBF[12]			**Truncated IBS**			**Blended IBS**		
Figure	*DIM*	CPU	SUR	*DIM*	CPU	SUR	*DIM*	CPU	SUR
Fig. 15(1)	*6,543*	15.94	36.61	*32,107*	2.17	2.81	*44,533*	0.84	3.32
Fig. 15(2)	*6,564*	12.04	35.54	*34,637*	2.04	3.14	*47,909*	0.73	4.51
Fig. 15(3)	*5,667*	10.25	35.63	*35,484*	2.21	2.25	*44,991*	0.72	4.08
Fig. 15(4)	*5,658*	11.49	38.30	*36,980*	2.18	4.25	*47,160*	0.86	3.79
Fig. 15(5)	*6,534*	12.20	40.41	*34,231*	1.97	3.61	*45,064*	0.78	2.46
Fig. 16(1)	*12,390*	23.87	210.08	*17,659*	4.94	6.92	*108,662*	9.10	10.66
Fig. 16(2)	*11,565*	31.80	219.26	*19,325*	4.24	7.11	*121,419*	9.73	10.31
Fig. 17	*7,071*	11.79	31.10	*37,124*	2.80	1.12	*33,605*	2.96	2.91

$$\Pi^{(d)}_{m,n} = \int_X B^{(d)}_m(x).B^{(d)}_n(x)dx \tag{32}$$

where $B^{(d)}$ refers to d-th derivative of B and $m,n \in \{0,1,...,N\}$. These matrices are $N \times N$, and we can compute the entries of $\mathbf{H}$ through them:

$$H_{k,l} = \Pi^{(2)}_{k_1,l_1}\Pi^{(0)}_{k_2,l_2} + 2\Pi^{(1)}_{k_1,l_1}\Pi^{(1)}_{k_2,l_2} + \Pi^{(0)}_{k_1,l_1}\Pi^{(2)}_{k_2,l_2} \tag{33}$$

Thus, in order to compute the $N^2 \times N^2$ matrix $\mathbf{H}$, we must compute three $N \times N$ matrices $\mathbf{\Pi}^{(0)}$, $\mathbf{\Pi}^{(1)}$ and $\mathbf{\Pi}^{(2)}$. Now the objective is to compute the entries of these matrices based on the blending function. For this purpose we consider three 4×4 matrices considering different combinations of the blending functions:

$$\pi^{(0)}_{r,s} = \int_0^1 b^{(0)}_r(u).b^{(0)}_s(u)du \tag{34}$$
$$\pi^{(1)}_{r,s} = \int_0^1 b'_r(u).b'_s(u)du$$
$$\pi^{(2)}_{r,s} = \int_0^1 b''_r(u).b''_s(u)du$$

where $b^{(d)}$ refers to d-th derivative of b and $r,s \in \{0,1,2,3\}$. Then, we start moving on the region, and accumulating those values of $\pi^{(0)}$, $\pi^{(1)}$ or $\pi^{(2)}$ that contribute in that sub-region. Note that we use a special property of the B-Spline basis functions: when they are considered in a sub-region (with the length of Δ) they are shrunk versions of one of the cubic patches.

REFERENCES

[1] M. Berger, J. Levine, L. Nonato, G. Taubin, and S. C.T., "A benchmark for surface reconstruction," *ACM Trans. on Graphics*, vol. 32, no. 2, p. 20, 2013.

[2] Y. Watanabe, T. Komuro, and M. Ishikawa, "High-resolution shape reconstruction from multiple range images based on simultaneous estimation of surface and motion," in *Proc. International Conference on Computer Vision*, 2009, pp. 1787–1794.

[3] G. Marola, "A technique for finding the symmetry axes of implicit polynomial curves under perspective projection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 465–470, 2005.

[4] H. Pottmann and S. Leopoldseder, "A concept for parametric surface fitting which avoids the parametrization problem," *Computer Aided Geometric Design*, vol. 20, no. 6, pp. 343–362, 2003.

[5] Q. Huang, B. Adams, and M. Wand, "Bayesian surface reconstruction via iterative scan alignment to an optimized prototype," in *Symposium on Geometry Processing*, 2007, pp. 213–223.

[6] B. Zheng, R. Ishikawa, T. Oishi, J. Takamatsu, and K. Ikeuchi, "A fast registration method using ip and its application to ultrasound image registration," in *IPSJ Transactions on Computer Vision and Applications*, September 2009, pp. 209–219.

[7] M. Rouhani and A. Sappa, "The richer representation the better registration," *IEEE Trans. on Image Processing*, vol. 22, no. 12, pp. 5036–5049, 2013.

[8] F. Mokhtarian and A. Mackworth, "A theory of multiscale, curvature-based shape representation for planar curves," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, no. 8, pp. 789–805, 1992.
[9] C. Oden, A. Ercil, and B. Buke, "Combining implicit polynomials and geometric features for hand recognition," *Pattern Recognition Letters* , vol. 24, no. 13, pp. 2145–2152, 2003.
[10] B. Zheng, J. Takamatsu, and K. Ikeuchi, "3d model segmentation and representation with implicit polynomials," *IEICE Transactions*, vol. 91-D, no. 4, pp. 1149–1158, 2008.
[11] G. Turk and J. O'Brien, "Shape transformation using variational implicit functions," *SIGGRAPH*, 1999.
[12] M. Blane, Z. Lei, H. Civil, and D. Cooper, "The 3L algorithm for fitting implicit polynomials curves and surface to data," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 3, pp. p 298–313, March 2000.
[13] http://shapes.aimatshape.net/, AIM@SHAPE, Digital Shape Work-Bench.
[14] J. Carr, R. Beatson, J. Cherrie, T. Mitchell, W. Fright, B. McCallum, and T. Evans, "Reconstruction and representation of 3d objects with radial basis functions," *SIGGRAPH*, pp. 67–76, 2001.
[15] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H. Seidel, "Multi-level partition of unity implicits," *ACM Trans. on Graphics*, vol. 22, no. 3, pp. 463–470, 2003.
[16] T. Sahin and M. Unel, "Fitting globally stabilized algebraic surfaces to range data," in *Proc. International Conference on Computer Vision*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 1083–1088.
[17] H. Dinh, G. Turk, and G. Slabaugh, "Reconstructing surfaces by volumetric regularization using radial basis functions," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 10, pp. 1358–1371, 2002.
[18] B. Morse, T. Yoo, D. Chen, P. Rheingans, and K. Subramanian, "Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions," in *Shape Modeling International*, 2001, pp. 89–98.
[19] M. Rouhani and A. Sappa, "Implicit b-spline fitting using the 3l algorithm," in *Proc. IEEE International Conference on Image Processing*, 2011, pp. 893–896.
[20] B. Jüttler and A. Felis, "Least-squares fitting of algebraic spline surfaces," *Adv. Comput. Math.*, vol. 17, no. 1-2, pp. 135–152, 2002.
[21] M. Aigner and B. Jutler, "Gauss-newton-type technique for robustly fitting implicit defined curves and surfaces to unorganized data points," *IEEE International Conference on Shape Modelling and Application*, pp. p 121–130, 2008.
[22] Z. Yang, J. Deng, and F. Chen, "Fitting unorganized point clouds with active implicit b-spline curves," *Visual Computing*, vol. 21, no. 1, pp. p 831–839, 2005.
[23] S. Ahn, W. Rauh, H. Cho, and H. Warnecke, "Orthogonal distance fitting of implicit curves and surfaces," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 620–638, May 2002.
[24] G. Taubin, "Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, no. 11, pp. 1115–1138, November 1991.
[25] T. Tasdizen, J. Tarel, and D. Cooper, "Improving the stability of algebraic curves for applications," *IEEE Trans. on Image Processing*, vol. 9, no. 3, pp. 405–416, 2000.
[26] A. Helzer and M. Barzohar, "Stable fitting of 2d curves and 3d surfaces by implicit polynomials," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 10, pp. 1283–1294, 2004, fellow-Malah, David.
[27] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and T. Silva, "Point set surfaces," in *IEEE Visualization*, 2001, pp. 21–28.
[28] B. Mederos, M. Laje, S. Arouca, F. Petronetto, L. Velho, T. Lewiner, and H. Lopes, "Regularized implicit surface reconstruction from points and normals," *J. Braz. Comp. Soc.*, vol. 13, no. 4, pp. 7–16, 2007.
[29] Y. Nagai, Y. Ohtake, and H. Suzuki, "Smoothing of partition of unity implicit surfaces for noise robust surface reconstruction," *Comput. Graph. Forum*, vol. 28, no. 5, pp. 1339–1348, 2009.
[30] D. Sharvit, J. Chan, H. Tek, and B. Kimia, "Symmetry-based indexing of image databases," *Journal of Visual Communication and Image Representation*, vol. 9, no. 4, pp. 366–380, 1998.
[31] D. Vlasic, I. Baran, W. Matusik, and J. Popovic, "Articulated mesh animation from multi-view silhouettes," *ACM Trans. on Graphics*, vol. 27, no. 3, 2008.
[32] K. Museth, D. Breen, R. Whitaker, and A. Barr, "Level set surface editing operators," *TOG*, vol. 21, no. 3, pp. 330–338, 2002.

Mohammad Rouhani (S'09) received the master degrees in applied mathematics from Sharif University of Technology, Tehran, Iran, in 2006. Having experienced two years of lecturing in Computer Science, he joined the Computer Vision Center in Barcelona, Spain, where he received the Ph.D. degree in Computer Science, in 2012. After holding a research position in ICVL Lab at Imperial College London, U.K., he joined Morpheo team at INRIA Grenoble. His current research interests include 3D Computer Vision and Computer Graphics, including shape modeling, surface reconstruction, deformation modeling, object registrations well as machine learning techniques for 3D pose estimation.

Angel Domingo Sappa (S'94-M'00) received the electromechanical engineering degree from National University of La Pampa, General Pico, Argentina, in 1995 and the Ph.D. degree in industrial engineering from the Polytechnic University of Catalonia, Barcelona, Spain, in 1999. In 2003, after holding research positions in France, the UK, and Greece, he joined the Computer Vision Center, where he is currently a senior researcher. He is a member of the Advanced Driver Assistance Systems Group. His research interests span a broad spectrum within the 2D and 3D image processing. His current research focuses on stereoimage processing and analysis, 3D modeling, and dense optical flow estimation. He is a member of the IEEE.

Edmond Boyer received the PhD degree from the Institut National Polytechnique de Lorraine, France, in 1996. He is an associate professor at Grenoble Universities, France. He started his professional career as a research assistant in the Department of Engineering, University of Cambridge, United Kingdom. He joined INRIA Grenoble in 1998. His fields of competence cover computer vision, computational geometry, and virtual reality. He is a cofounder of the 4D View Solution Company in the domain of spatiotemporal modeling. His current research interests are in 3D dynamic modeling from images and videos, motion capture and recognition from videos, and immersive and interactive environments.